

**Overview**

ofsde processes real time data from the IHFS database and outputs files which are read by CHPS/FEWS. Types of processing performed by ofsde include transforming PC data into PP values, changing two character SHEF pe types into OFS types, station id translation based on SHEF source code and checking end times against a window around 12z for PP24 data. It is normally run via the cron between 2 and 6 times per hour.

**Latest Changes**

- removed RC24 generation using the mape function (FB 1949)  
this removes the problem resulting in the error message that the  
.../rfc/nwsrfs/ofs/files/oper/fs5files/USERPARM  
file is not found
- added processing of Water Quality data (FB 1672)
- tested ofsde using postgres version 9.3.5 (FB 1615)
- added reference to new CHPS/ofsde doc describing installation on a CHPS system (FB 1478)
- added check for multiple ts codes for PC data processing AND added additional checks for PP06 records in CurPP table to determine if write to vl2v5.chps is necessary (FB 1364)

**Outline**

- (1) Open log file and vl2v5.chps file
- (2) Read parameters from .Apps\_defaults file
- (3) Open database
- (6) Query the PerfLog table to find the last ofsde run time
- (7) Get all records from CurPC table in linked list form with posting time later than last ofsde run time  
(records are ordered by lid,ts,datetime)

For all records found:

- (a) Check IngestFilter table for ofs\_input value
- (b) If ofs\_input = "T" then

```

        (c) Get OFS data type and forward/backward
            window from the OFSDataTrans table
        (d) If record has same id as previous record AND
            obstime is within same 12z - 12z period as
            previous record, then
            (e) Take next record
        Else
            (f) Process record
        End if

    End If

(8) Get all records from CurPP table in linked list form with
    posting time later than last ofsde run time
    (records are ordered by lid,ts,datetime)

    For all records found:
        (a) Check IngestFilter table for ofs_input value

        (b) If ofs_input = "T" then
            (b1) if value in CurPP table not equal to -9999.
            and check shef revision flag
            (b2) Get OFS data type and forward/backward
            window from the OFSDataTrans table
            (b3) Process record
        End If

(9) Get all records from Temperature table in linked list form
    with posting time later than last ofsde run time

    For all records found:
        (a) Check IngestFilter table for ofs_input value
        (b) If ofs_input = "T" then

            (c) Get OFS data type from OFSDataTrans table
            (d) Process record

        End if

(10) Get all records from Height table in linked list form with
    posting time later than last ofsde run time
    (same for Discharge, Snow, Ice, Moisture, Lake, Ground,
    GateDam tables)

```

For all records found:

- (a) Check IngestFilter table for ofs\_input value
- (b) If ofs\_input = "T" then
  - (c) Get OFS data type from OFSDataTrans table
  - (d) Process record

End if

- (11) Get all records from Evaporation table in linked list form with posting time later than last ofsde run time (same for Radiation, Agricultural, YUnique, WaterQuality tables)

For all records found:

- (a) Check IngestFilter table for ofs\_input value
- (b) If ofs\_input = "T" then
  - (c) Get OFS data type from OFSDataTrans table
  - (d) Process record

End if

- (12) Get all records from ProcValue table in linked list form with posting time later than last ofsde run time (records in the ProcValue table have SHEF type ="P")

For all records found:

- (a) Check IngestFilter table for ofs\_input value
- (b) If ofs\_input = "T" then
  - (c) Get OFS data type from OFSDataTrans table
  - (d) Process record

End if

- (13) Get all records from FcstDischarge table in linked list form with posting time later than last ofsde run time

For all records found:

- (a) Check IngestFilter table for ofs\_input value
- (b) If ofs\_input = "T" then
  - (c) If there are multiple records with the same

ts and validtime, then choose the record with the most recent basistime (i.e. the most recently created forecast value)  
(d) Get OFS data type from OFSDataTrans table  
(e) Process record

End if

(14) Get all records from FcstHeight table in linked list form with posting time later than last ofsde run time

For all records found:

(a) Check IngestFilter table for ofs\_input value  
(b) If ofs\_input = "T" then  
  
(c) If record has same id,pe,validtime as previous record, then read next record  
(d) Get OFS data type from OFSDataTrans table  
(e) Process record

End if

(15) Get all records from FcstTemp table in linked list form with posting time later than last ofsde run time

For all records found:

(a) Check IngestFilter table for ofs\_input value  
(b) If ofs\_input = "T" then  
  
(c) If record has same id,extremum,validtime as previous record, then read next record  
(d) Process record

End if

(16) Write number of records processed, time of run to Perflog table

(17) Close database and exit

Notes

(1) List of data types processed by ofsde:

Observed Data:

Precipitation (both PP and PC data)  
Temperature  
Height  
Discharge  
Snow  
Ice  
Moisture  
Lake  
Ground  
Radiation  
Evaporation  
Agricultural  
Unique data for sites (stored in the YUnique table)  
Water Quality  
GateDam

Forecast Data:

Discharge  
Height  
Temperature

(2) No processing is done for the following data types:

Forecast Precip  
Forecast Other  
Fish Count  
Power  
Weather  
Wind

Data for these data types is not written to the ofsde  
output files.

**Subprocesses**

(1) qcrng

- range checks on data using DataLimits and LocDataLimits tables
- if station id does not appear in the LocDataLimits table  
AND the pe type does not appear in the DataLimits  
table, then data passes qc check
- used only for PP data generated from PC data

(2) pc2pp

- calculate PP06 and PP24 data at sub synoptic times

- call qcrng
  - call wr2ofs
- (4) wr2ofs
    - station identifier translation using OFSStationTrans table
    - PE translation from SHEF to V5 form using OFSDataTrans table
    - write records to vl2v5.chps file
  - (5) rrs2ofs
    - same as wr2ofs except writes duration code and 'FUT' for future data
  - (6) durdec
    - decodes the SHEF duration code into hours
  - (7) chktatm
    - checks obs time of observed TA data for possible change to 12z
  - (8) chkpostpp
    - checks obs time of observed PP data for possible posting to OFS and possible change of obs time to 12z
  - (9) process observed precip records
    - for PC data:
      - call pc2pp
    - for PP data:
      - if quality\_code = F, then ignore data
      - if ofsde\_check\_questionable token = ON and quality\_code = Q, then ignore data
      - if duration = 2001 or 1024 or 5004 then call chkpostpp
      - call wr2ofs
      - end if
  - (10) process observed height, discharge, snow, ice, moisture, lake, ground (RRS data) records
    - if quality\_code = F, then ignore data
    - if ofsde\_check\_questionable token = ON and quality\_code = Q, then ignore data
    - change values = -9999. to -999.
    - if ofsde\_rrstime\_check = ON AND obs time is not 12z AND

- forward window and back window not both = 0.0 then
  - call chkrrstime
  - call durdec
  - call rrs2ofs
- (11) process observed temperature records
  - if quality\_code = F, then ignore data
  - if ofsde\_check\_questionable token = ON and quality\_code = Q, then ignore data
  - for max/min temperatures, check that min < max and max > min
  - call chktatm
  - call wr2ofs
- (12) process observed radiation, evaporation, agricultural, water quality, unique records and processed data (from ProcValue table)
  - if quality\_code = F, then ignore data
  - if ofsde\_check\_questionable token = ON and quality\_code = Q, then ignore data
  - change values = -9999. to -999.
  - if pe = "SA" and value != -999. then divide value by 100. (SHEF pe = "SA" corresponds to OFS pe = "AESC")
  - call wr2ofs
- (13) process forecast discharge records
  - call durdec
  - call rrs2ofs
- (14) process forecast height records
  - call durdec
  - call rrs2ofs
- (15) process forecast temperature records
  - PE translation from SHEF to V5 form using OFSDataTrans table
  - call wr2ofs

### **qcrng Process**

- (1) Check the data value against the max and min values from the LocDataLimits table for the station id, pe type and date
- (2) if value is between max and min then

```

        return status=passed qc test

    else if value is outside of range, then
        return status=failed qc test

    else if no record is found in the table, then check value
        against max/min values from the DataLimits table for
        pe type and date

(3) if value is between max and min then
    return status=passed qc test
else if value is outside of range, then
    return status=failed qc test

(4) if no record is found in either table, then
    return status=passed qc test

```

### **pc2pp Process**

- (1) Read PC values for given id and ts for 12z to 12z period of interest (subroutine rd12bf) (see notes 1,4,5)
- (2) Find the 5 data values with obs date/time within allowable window around the sub-synoptic times  
If no value is available within allowable window, then mark PC value as missing
- (3) Attempt to estimate missing sub-synoptic PC values (subroutine estpc)  
If a PC value is missing, then
  - (a) find closest earlier and closest later PC values which bracket (in time) the missing PC value
  - (b) if either value is not available, then go to (4)
  - (c) if the two values are within 0.05 in absolute value, then set the missing value to the closest later PC value
 End If
- (4) Check the Ingestfilter table for PP duration = 1006  
If not found, then go to (10)
- (5) Calculate 6 hour period totals



- (6) Check CurPP table for a record with the same id, ts, obstime
  - (A) If a record is found, then
    - If value  $\geq 0.0$  OR (value  $< 0.0$  AND revision code = 1)
      - then go to step(9)
      - (do not write data to vl2v5.chps file)
    - End If
    - Else
      - Write record to vl2v5.chps file
    - End If
- (7) Execute subroutine qcrng
  - (A) If data fails qc test, then set value = -999.
- (8) Execute subroutine wr2ofs
- (9) Repeat steps (6), (7) and (8) for the three remaining periods
- (10) Calculate 12z-12z total or partial day total
  - For partial day total:
    - if the PC value for the earlier 12z time is missing, then
      - PP24 value is not calculated
    - else
      - if number of additional sub-synoptic PC values is  $> 0$ 
        - then
          - if the latest sub-synoptic time of the PC value is  $> 18z$ , then
            - partial day total = PC(latest obs time) - PC(previous 12z)
          - else
            - PP24 value set to missing
        - else
          - PP24 value set to missing
- (11) Check CurPP table for a record with the same id, ts code and obstime
  - (A) If found, then go to step (14) - do not write data to vl2v5.chps file
- (12) Execute subroutine qcrng
  - (A) If data fails qc test, then set value = -999.
- (13) Execute subroutine wr2ofs

(14) Exit

\*\*\*\*\*  
\*

#### Notes

- (1) Step (1) also saves PC values before the earlier 12z time and after the later 12z time for use in estimating missing PC values in step (3).
- (2) PP06 values are assigned times = 000000,60000,120000,180000.
- (3) The allowable window on obs time is read from the OFSDataTrans table. Separate values for forward in time and back in time windows are stored.
- (4) Step (1) reads through the PC records in the 12z - 12z time period twice. The first time to count the number of records for a malloc and the second time to actually store the values. To accomplish this, records are selected from the database ONCE, placed into a linked list and the list is traversed twice.
- (5) Added test for case where obstime is AFTER 12z but within allowable window around 12z. This is considered to be the end of the 12z-12z period. (FB 1320)

#### **Processing Forecast Temperature Data**

ndate = value of .Apps\_defaults token ofsde\_ndate  
if not defined, default value = 7 is used

maxfuttime = current time + ndate days

- (1) search the FcstTemp table for records with validtime > current time AND validtime < maxfuttime
  - (2) If found, then
    - (A) Select records with newest basistime for each id, pe, ts, ex, validtime
    - (B) call rrs2ofs
- End If

#### **Processing Forecast Height and Discharge Data**

*Version: OHD-Core-CHPS-4.5.a*

*Release Date: 28 October 2017*

In step (1) below, the height data is selected from the fcstheight table while the discharge data is read from the fcstdischarge table.

(1) Search the table for records with validtime > current time

(2) If found, then

(A) Select records with newest basistime for each id, pe,  
ts, validtime

(B) call durdec

(C) If SHEF duration code is successfully transformed into  
hours, then call rrs2ofs

End If

(3) Exit

## **CHKPOSTPP**

This process checks the obs time and duration code of PP data to determine if the data should be posted to FEWS. It also checks the obs time of the PP data being posted to FEWS to determine if a change to the obs time is necessary. See documentation in NWSRFS Users Manual Chap VI p 4.2-5. This check is done for duration codes = 1024 and 2001 only.

iotime = observation time (hhmmss) (z time)

idur = SHEF duration code

ofsde\_lower\_ppp, ofsde\_upper\_ppp = .Apps\_defaults tokens

default values: ofsde\_lower\_ppp = 2

ofsde\_upper\_ppp = 2

iobl = 120000 - (ofsde\_lower\_ppp\*10000)

iobu = 120000 + (ofsde\_upper\_ppp\*10000)

if(iotime < iobl OR iotime > iobu) AND (idur = 2001 OR idur =  
1024)

set flag to not post data to FEWS and return

```

else

    set flag to post data to FEWS

end if

if(iotime >= iobl AND iotime <= iobu)

    iotime changed to 120000

end if

```

### **CHKTATM Process**

This process checks the obs time of max/min temperatures to determine if a change to the obs time is necessary. This is done to satisfy requirements from OFS. See documentation in NWSRFS Users Manual Chap VI p 4.2-5.

```

shs = SHEF source code
shex = SHEF extremum code
iotime = observation time (hhmmss) (z time)
intlXXX, intuXXX = .Apps_defaults tokens

```

```

default values: intlrmn = 8
                  inturmn = 2
                  intlrmn = 2
                  inturzn = 2
                  intlrmn = 8
                  inturzn = 2

```

```

if shs = M then

    if shex = N then

        iobl = 12 - intlrmn
        iobu = 12 + inturmn

        if(iotime >= iobl AND iotime <= iobu) then iotime changed
to 12z

    else if shex = X then

```

```

        iobl = 12 - intlrzx
        iobu = 12 + inturzx

        if(iotime >= iobl AND iotime <= iobu) then iotime changed
to 12z

        end if

else if shs = Z then

        if shex = N OR shex = P then

                iobl = 12 - intlrzn
                iobu = 12 + inturzn

                if(iotime >= iobl AND iotime <= iobu) then iotime changed
to 12z

        else if shex = X then

                iobl = 12 - intlrzx
                iobu = 12 + inturzx

```

### **CHKRRSTIME Process**

This process was added at the request of NWRFC to check obs times of observed RRS data against a window around 12z. Originally, NWRFC requested this time check for Discharge data only. During testing, it was decided to generalize it for all RRS data types. If the obs time is not 12z but is within the window, then the obs time is changed to 12z before the record is written to vl2v5.chps. The window is defined by reading the fwd\_time\_window and bkw\_time\_window fields from the OFSDataTrans table. Units of these values is hours.

```

iotime = observation time (hhmmss) (z time)
ifwin = integer portion of forward window
ibwin = integer portion of backward window

idfwin = decimal portion of forward window
idbwin = decimal portion of backward window

```

```
iobf= 120000 + (ifwin*10000) + (idfwin*6000)
iobb= 120000 - (ibwin*10000) - (idbwin*6000)
```

```
if (iotime >= iobb AND iotime <= iobf) then change iotime to 12z
```

### **DURDEC Process**

The durdec process decodes the duration code into a duration in hours and in some cases changes the hour of observation.

The following is an outline describing the procedure:

Input: SHEF duration code, hour of observation (=hrobs), windows around observation time (=ofsde\_lower\_ppp and ofsde\_upper\_ppp)

Output: duration in hours (=durhr), hour of observation (=hrobs)

- (1) If SHEF duration code = 0, then
  - (a) durhr = 0
  - (b) hrobs unchanged
  - (c) returnEnd If
- (2) Decode duration code into first digit (=type) and last two digits (=nhrs)
- (3) If type = 1, then
  - (a) durhr = nhrs
  - (b) hrobs unchangedElse If type = 2, then
  - (c) durhr = (nhrs \* 24)
  - (d) if hrobs >= 12, then
    - (i) iw = hrobs -12
    - (ii) int = ofsde\_lower\_ppp
  - else
    - (iii) iw = 12 - hrobs
    - (iv) int = ofsde\_upper\_ppp
  - end if
  - (e) if iw <= int, then hrobs = 12Else If type = 5, then
  - (f) if hrobs >= 12, then

```

        (i) iw = hrobs - 12
        (ii) int = ofsde_lower_ppp
    else
        (iii) iw = 12 - hrobs
        (iv) int = ofsde_upper_ppp
    end if
(g) if iw <= int, then
    (i) durhr = 24
    (ii) hrobs = 12
else
    (iii) durhr = hrobs - 12
    (iv) if durhr < 0, then durhr = hrobs + 12
end if
Else SHEF duration code cannot be transformed into hours
End If

```

#### Notes

(1) ofsde\_lower\_ppp and ofsde\_upper\_ppp are currently read from .Apps\_defaults.

#### **mape Process**

Previous versions of ofsde generated RC24 data via the mape function. This function read the NWSRFS USERPARM data.

The mape function was removed from the ofsde executable in Feb 2016. (FB 1949)

#### **vl2v5.chps File Format**

The vl2v5.chps file is an ascii file which is output from ofsde, read by the chpspst script and copied to the FEWS import directory. In August 2012, it was determined that a sort is required for proper ingest into FEWS. See Fogbugz 934 for further info. The vl2v5.chps file is generated by each run of ofsde.

<u>Field Description</u>	<u>Format</u>
station identifier	A8
date (yyyymmdd)	I8
time (hhmm)	I4.4

PE code for OFS	A4
type/source code	A2
value	F11.2

Following fields are written for RRS data only:

future data indicator	A3
duration (hours)	I2

#### Notes

- (1) There is no space between the date and time fields.  
All other fields are separated by one space.
- (2) Values greater than 1e8 are checked for and flagged as errors. This prevents the case of the value field containing \*\*\*\*\* as is the case when the format is exceeded.
- (3) The future data indicator field = "FUT" for future RRS data and blank for observed RRS data.
- (4) The duration is zero for instantaneous data.

#### **.Apps\_defaults tokens**

```
ofsde_log_dir : # dir containing vl2v5.chps file
               # and log files

ofsde_lower_ppp      : 2  # number of hours before 12z for
                          # changing PPP obstime to 12z
                          # number of hours before 12z for
                          # posting 1024 and 2001 data

ofsde_upper_ppp      : 2  # number of hours after 12z for
                          # changing PPP obstime to 12z
                          # number of hours after 12z for
                          # posting 1024 and 2001 data

ofsde_ndate    : 7      # number of days to search for
```



```

# forecast temperature data

ofsde_rrstime_check    : ON  # check obs time of observed RRS
                          # data against window around 12z

ofsde_check_questionable : ON  # ON/OFF token for checking
                              # for records with
                              # quality_code = Q
                              # if token = ON and code = Q
                              # then record is ignored

# intervals for max/min temperatures
# these represent number of hours around 12z
# naming scheme: intXYZ
                        X = 1 - lower interval
                        u - upper interval

                        YY = SHEF type/source = RM or RZ

                        Z = SHEF extremum code = X or N

intlrmn                : 8
inturmn                : 2
intlrrzn               : 2
inturzn               : 2
intlrrzx               : 8
inturzx               : 2

```

### Forecast Discharge Processing Note

After fixing a problem processing the RQOT data described in FogBugz 826, it was discovered that the output from ofsde was not being properly ingested by CHPS. The problem was found to be that CHPS requires that its input data be sorted. The following ofsde output is from an NCRFC case for RQOT data:

```

SVRM7    201208241100 RQOT FZ    33500.00 FUT 0
SVRM7MVS 201208241100 RQOT FC    33500.00 FUT 0
SVRM7    201208251100 RQOT FZ    31000.00 FUT 0
SVRM7MVS 201208251100 RQOT FC    31000.00 FUT 0
SVRM7    201208261100 RQOT FZ    30000.00 FUT 0
SVRM7MVS 201208261100 RQOT FC    30000.00 FUT 0
SVRM7    201208271100 RQOT FZ    29500.00 FUT 0
SVRM7MVS 201208271100 RQOT FC    29500.00 FUT 0
SVRM7    201208281100 RQOT FZ    29000.00 FUT 0
SVRM7MVS 201208281100 RQOT FC    29000.00 FUT 0
SVRM7    201208291100 RQOT FZ    28500.00 FUT 0
SVRM7MVS 201208291100 RQOT FC    28500.00 FUT 0

```

The output above needs to be sorted as follows to allow it to be ingested by FEWS:

```
SVRM7      201208241100 RQOT FZ      33500.00 FUT 0
SVRM7      201208251100 RQOT FZ      31000.00 FUT 0
SVRM7      201208261100 RQOT FZ      30000.00 FUT 0
SVRM7      201208271100 RQOT FZ      29500.00 FUT 0
SVRM7      201208281100 RQOT FZ      29000.00 FUT 0
SVRM7      201208291100 RQOT FZ      28500.00 FUT 0
SVRM7MVS   201208241100 RQOT FC      33500.00 FUT 0
SVRM7MVS   201208251100 RQOT FC      31000.00 FUT 0
SVRM7MVS   201208261100 RQOT FC      30000.00 FUT 0
SVRM7MVS   201208271100 RQOT FC      29500.00 FUT 0
SVRM7MVS   201208281100 RQOT FC      29000.00 FUT 0
SVRM7MVS   201208291100 RQOT FC      28500.00 FUT 0
```

The sort command used to generate the output above is

```
sort -k 1,1 -k 3,4 -k 2,2
```

which sorts first on the 1st column (ID), then on the 3rd and 4th columns (pe code and type/source) and finally on the 2nd column (date/time).

## **Description of ofsde output Files**

(1) ofsde log files:

```
/.../ofsde.mmddyyyyhhmm
```

(2) ofsde output imported to FEWS:

```
/.../vl2v5.chps
```

FEWS uses its own configuration file to define the directory to use for importing the vl2v5.chps file. The setting is defined like the following

```
IMPORT_FOLDER_OFSDE=$IMPORT_FOLDER_ROOT$/ofsde
```

Note that \$IMPORT\_FOLDER\_ROOT\$ is RFC dependent, but usually something like /awips/chps\_local/data/toCHPS/\$RFCNAME. This directory

should be a directory local to the machines chps3-nhdr, chps6-nhdr and chps9-nhdr.

(3) See the doc Installing the CHPS Ofsde for information on directory locations for the scripts and executable necessary for running ofsde at a CHPS site.

### Testing at OWP

At OWP, there is a real-time data feed for NERFC (TAR) set up for testing. This test is running against a db using postgres 9.3.5. There is also a regression test. This test is running against db using postgres 9.2.9.

For FB 1615, ofsde was tested against postgres 9.3.5 and found to work properly.

FB 2080 was written to describe the testing on the nmtr machine.

### Sample ofsde Output

vl2v5.chps sample:

ABTM1	201410071400	PP01	RG	0.00	
ABTM1	201410071500	PP01	RG	0.00	
ABTM1	201410071600	PP01	RG	0.00	
ABTM1	201410061800	PP06	RG	0.00	
ABTM1	201410070000	PP06	RG	0.00	
ABTM1	201410070600	PP06	RG	0.00	
ABTM1	201410071200	PP06	RG	0.00	
ABTM1	201410071200	PP24	RG	0.00	
BBOV1	201410061800	PP06	R2	0.00	
BBOV1	201410070000	PP06	R2	0.00	
BBOV1	201410070600	PP06	R2	0.00	
BBOV1	201410071200	PP06	R2	0.00	
BBOV1	201410061800	PP06	RG	0.00	
BBOV1	201410070000	PP06	RG	0.00	
BBOV1	201410070600	PP06	RG	0.00	
BBOV1	201410071200	PP06	RG	0.00	
BBOV1	201410071200	PP24	R2	0.00	
BBOV1	201410071200	PP24	RG	0.00	
BBOV1	201410070545	TA03	RG	46.73	
BBOV1	201410070550	TA03	RG	46.55	
YTCC3	201410071400	STG	RG	1.15	0
YTCC3	201410071415	STG	RG	1.15	0

YTCC3	201410071430	STG	RG	1.14	0
SVRM7	201208271100	RQOT	FZ	29500.00	FUT 0
SVRM7	201208281100	RQOT	FZ	29000.00	FUT 0
SVRM7	201208291100	RQOT	FZ	28500.00	FUT 0

### Sample of ofsde log:

version number = OHD-CORE-CHPS-4.1.a  
hostname: ingest3

intlppp = 2 intuppp = 2 ndate = 7  
ProcValue table search  
database name = hd\_ob92rha  
last ofsde run time = 2002-04-09 13:20:01 Z

processing CurPC table records

```
record = ALTP1 PC 0 RG Z 2002-04-09 13:00:00 10.970000 Z 1879048191
  57 records read from OFSDataTrans table
    n25=2 nb=2 nf=0
    value=10.970000 datetime=2002-04-09 12:00:00
    value=10.970000 datetime=2002-04-09 13:00:00
    BEFORE ESTPC: 04-09 12 = 10.97 04-09 18 = 999.00 04-10 00 = 999.00 04-10 06 =
= 999.00 04-10 12 = 999.00
    AFTER ESTPC: 04-09 12 = 10.97 04-09 18 = 999.00 04-10 00 = 999.00 04-10 06 =
= 999.00 04-10 12 = 999.00
    1 NON MISSING values FOUND -- NO PP PROCESSING DONE
record = AXMP1 PC 0 RG Z 2002-04-09 13:00:00 45.240000 Z 1879048191
  n25=2 nb=2 nf=0
  value=45.240000 datetime=2002-04-09 12:00:00
  value=45.240000 datetime=2002-04-09 13:00:00
  BEFORE ESTPC: 04-09 12 = 45.24 04-09 18 = 999.00 04-10 00 = 999.00 04-10 06 =
= 999.00 04-10 12 = 999.00
  AFTER ESTPC: 04-09 12 = 45.24 04-09 18 = 999.00 04-10 00 = 999.00 04-10 06 =
= 999.00 04-10 12 = 999.00
  1 NON MISSING values FOUND -- NO PP PROCESSING DONE
record = BBRP1 PC 0 RG Z 2002-04-09 13:00:00 6.080000 Z 1879048191
  n25=2 nb=2 nf=0
  value=6.080000 datetime=2002-04-09 12:00:00
  value=6.080000 datetime=2002-04-09 13:00:00
  BEFORE ESTPC: 04-09 12 = 6.08 04-09 18 = 999.00 04-10 00 = 999.00 04-10 06 =
999.00 04-10 12 = 999.00
  AFTER ESTPC: 04-09 12 = 6.08 04-09 18 = 999.00 04-10 00 = 999.00 04-10 06 =
999.00 04-10 12 = 999.00
  1 NON MISSING values FOUND -- NO PP PROCESSING DONE
record = BHLPl PC 0 RG Z 2002-04-09 12:34:00 3.180000 Z 1879048191
  INSUFFICIENT DATA IN CurPC table TO CALCULATE PPXX values
record = BMGP1 PC 0 RG Z 2002-04-09 12:30:00 0.150000 Z 1879048191
  n25=3 nb=4 nf=0
  value=0.150000 datetime=2002-04-09 12:00:00
  value=0.150000 datetime=2002-04-09 12:30:00
  value=0.150000 datetime=2002-04-09 13:00:00
  BEFORE ESTPC: 04-09 12 = 0.15 04-09 18 = 999.00 04-10 00 = 999.00 04-10 06 =
999.00 04-10 12 = 999.00
```

AFTER ESTPC: 04-09 12 = 0.15 04-09 18 = 999.00 04-10 00 = 999.00 04-10 06 = 999.00 04-10 12 = 999.00

1 NON MISSING values FOUND -- NO PP PROCESSING DONE

record = BMGP1 PC 0 RG Z 2002-04-09 13:00:00 0.150000 Z 1879048191

AFTER ESTPC: 04-09 12 = 16.17 04-09 18 = 999.00 04-10 00 = 999.00 04-10 06 = 999.00 04-10 12 = 999.00

1 NON MISSING values FOUND -- NO PP PROCESSING DONE

\*\*\*\*\* PP data processing \*\*\*\*\*

record = PETW2 PP 5004 RZ Z 2002-04-09 11:00:00 0.000000 Z 1879048191

obs time changed to 12z

WRITING DATA TO VL2V5.BUF --- PETW2 200204091200 PP24 0.00

processing Height table records

record = LNCPl HG 0 RG Z 2002-04-09 12:15:00 3.030000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- LNCPl 200204091215 STG 3.03  
record = LNCPl HG 0 RG Z 2002-04-09 12:30:00 3.030000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- LNCPl 200204091230 STG 3.03  
record = LNCPl HG 0 RG Z 2002-04-09 12:45:00 3.030000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- LNCPl 200204091245 STG 3.03  
record = LNCPl HG 0 RG Z 2002-04-09 13:00:00 3.020000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- LNCPl 200204091300 STG 3.02  
record = HGSP1 HG 0 RG Z 2002-04-09 09:15:00 1.570000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- HGSP1 200204090915 STG 1.57  
record = ILTP1 HG 0 RG Z 2002-04-09 12:30:00 3.650000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- ILTP1 200204091230 STG 3.65  
record = ILTP1 HG 0 RG Z 2002-04-09 12:45:00 3.650000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- ILTP1 200204091245 STG 3.65  
record = TNKP1 HG 0 RG Z 2002-04-09 10:00:00 2.330000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- TNKP1 200204091000 STG 2.33  
record = TNKP1 HG 0 RG Z 2002-04-09 10:15:00 2.330000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- TNKP1 200204091015 STG 2.33  
record = TNKP1 HG 0 RG Z 2002-04-09 10:30:00 2.330000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- TNKP1 200204091030 STG 2.33

processing Temperature table records

record = WFXPl TA 0 RG Z 2002-04-09 13:00:00 59.000000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- WFXPl 200204091300 TA03 59.00  
record = LNCPl TA 0 RG Z 2002-04-09 12:15:00 61.000000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- LNCPl 200204091215 TA03 61.00  
record = ILTP1 TA 0 RG Z 2002-04-09 12:45:00 61.000000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- ILTP1 200204091245 TA03 61.00  
record = ILTP1 TA 0 RG Z 2002-04-09 13:00:00 61.000000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- ILTP1 200204091300 TA03 61.00  
record = WIBPl TA 0 RG Z 2002-04-09 12:15:00 64.000000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- WIBPl 200204091215 TA03 64.00  
record = WIBPl TA 0 RG Z 2002-04-09 12:30:00 64.000000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- WIBPl 200204091230 TA03 64.00  
record = WIBPl TA 0 RG Z 2002-04-09 12:45:00 64.000000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- WIBPl 200204091245 TA03 64.00  
record = WIBPl TA 0 RG Z 2002-04-09 13:00:00 64.000000 Z 1879048191  
WRITING DATA TO VL2V5.BUF --- WIBPl 200204091300 TA03 64.00  
record = PETW2 TA 0 RZ X 2002-04-09 11:00:00 76.000000 Z 1879048191  
obs time changed to 12z  
WRITING DATA TO VL2V5.BUF --- PETW2 200204091200 TX24 76.00  
record = PETW2 TA 0 RZ N 2002-04-09 11:00:00 50.000000 Z 1879048191  
obs time changed to 12z  
WRITING DATA TO VL2V5.BUF --- PETW2 200204091200 TN24 50.00

processing WaterQuality table records

```
record = PHXN6 WV 0 RG Z 2015-06-09 13:15:00 1.660000 Z 1879048191
      WRITING DATA TO VL2V5.BUF --- PHXN6 201506091315 VIN 1.66
record = PHXN6 WV 0 RG Z 2015-06-09 13:30:00 1.540000 Z 1879048191
      WRITING DATA TO VL2V5.BUF --- PHXN6 201506091330 VIN 1.54
record = PHXN6 WV 0 RG Z 2015-06-09 13:45:00 1.550000 Z 1879048191
      WRITING DATA TO VL2V5.BUF --- PHXN6 201506091345 VIN 1.55
record = PHXN6 WV 0 RG Z 2015-06-09 14:00:00 1.630000 Z 1879048191
      WRITING DATA TO VL2V5.BUF --- PHXN6 201506091400 VIN 1.63
record = HBRN6 WV 0 RG Z 2015-06-09 13:30:00 0.580000 Z 1879048191
      WRITING DATA TO VL2V5.BUF --- HBRN6 201506091330 VIN 0.58
record = HBRN6 WV 0 RG Z 2015-06-09 13:45:00 0.580000 Z 1879048191
      WRITING DATA TO VL2V5.BUF --- HBRN6 201506091345 VIN 0.58
record = HBRN6 WV 0 RG Z 2015-06-09 14:00:00 0.550000 Z 1879048191
      WRITING DATA TO VL2V5.BUF --- HBRN6 201506091400 VIN 0.55
record = HBRN6 WV 0 RG Z 2015-06-09 14:15:00 0.620000 Z 1879048191
      WRITING DATA TO VL2V5.BUF --- HBRN6 201506091415 VIN 0.62
record = BYRN6 WV 0 RG Z 2015-06-09 13:45:00 1.470000 Z 1879048191
      WRITING DATA TO VL2V5.BUF --- BYRN6 201506091345 VIN 1.47
```

processing FcstHeight table records

```
record = RMDV2 HG 0 FF Z -1.0000 2002-04-09 18:00:00 2002-04-09 13:27:00 4.400000 Z
1879048191
      WRITING DATA TO VL2V5.BUF --- RMDV2 200204091800 STG 4.40
record = RMDV2 HG 0 FF Z -1.0000 2002-04-10 00:00:00 2002-04-09 13:27:00 4.300000 Z
1879048191
      WRITING DATA TO VL2V5.BUF --- RMDV2 200204100000 STG 4.30
record = RMDV2 HG 0 FF Z -1.0000 2002-04-10 06:00:00 2002-04-09 13:27:00 4.400000 Z
1879048191
      WRITING DATA TO VL2V5.BUF --- RMDV2 200204100600 STG 4.40
record = RMDV2 HG 0 FF Z -1.0000 2002-04-10 12:00:00 2002-04-09 13:27:00 4.400000 Z
1879048191
      WRITING DATA TO VL2V5.BUF --- RMDV2 200204101200 STG 4.40
record = RMDV2 HG 0 FF Z -1.0000 2002-04-12 00:00:00 2002-04-09 13:27:00 4.400000 Z
1879048191
      WRITING DATA TO VL2V5.BUF --- RMDV2 200204120000 STG 4.40
record = RMDV2 HG 0 FF Z -1.0000 2002-04-12 06:00:00 2002-04-09 13:27:00 4.400000 Z
1879048191
      WRITING DATA TO VL2V5.BUF --- RMDV2 200204120600 STG 4.40
record = RMDV2 HG 0 FF Z -1.0000 2002-04-12 12:00:00 2002-04-09 13:27:00 4.600000 Z
1879048191
      WRITING DATA TO VL2V5.BUF --- RMDV2 200204121200 STG 4.60
```

177 records written to vl2v5.buf file

CPU TIME USED= 0.110000 SEC -- ELAPSED TIME= 13.000000 SEC

**DFD for ofsde**

